Remarks

Reconsideration of the application is respectfully requested in view of the foregoing amendments and following remarks. Claims 1-13, 15-23, and 25-55 are pending in the application. No claims have been allowed. Claims 1, 15, 32, 44, 48, 50, and 53 are independent. Claims 1, 2, 5, 9, 11, 13, 15, 16-18, 23, 25, 27, 29, 32, 44, 46, 47, 48, and 50 have been amended. Claims 14 and 24 have been canceled without disclaimer or prejudice to renewal. Claims 54 and 55 have been added.

Independent claim 53 has not been amended. Accordingly, "a second or any subsequent action on the merits in any application or patent undergoing reexamination proceedings will not be made final if it includes a rejection, on newly cited art, other than information submitted in an information disclosure statement filed under 37 CFR 1.97(c) with the fee set forth in 37 CFR 1.17 (p), of any claim not amended by applicant or patent owner in spite of the fact that other claims may have been amended to require newly cited art." MPEP § 706.07(a).

IDS References Not Considered

Applicants submitted a number of IDS references in accordance with 37 CFR 1.97 and 1.98. However, while the majority of the IDS references were considered, two of the IDS references apparently were not considered (they were not initialed by the Examiner). The following is a list of the two IDS references that were apparently not considered:

IDS Dated	**Reference**
7/14/2004	"Scale Download"; Dept. of Computer Science, University of Massachusetts Amherst; http://www-ali.cs.umass.edu/Scale/download.html visited on May 20, 2003; pp 1-13
7/14/2004	"Scale"; Dept. of Computer Science, University of Massachusetts Amherst; http://www-ali.cs.umass.edu/Scale/ visited on May 20, 2003; pp 1-46

Applicants submit herewith Exhibit A, a postcard indicating that the references were received. Applicants note that the references are available in the Image File Wrapper, but the quality of the scanned images appears to be poor. For the convenience of the Examiner, Applicants enclose another copy of the references, which should produce a better quality scanned

image, and a 1449 form listing them (note that while the content of the references is the same, the number of pages has changed due to the print format).

Applicants respectfully request that the Examiner initial the appropriate boxes to indicate consideration of the two above-referenced IDS references that were submitted in accordance with procedure specified by 37 CFR 1.97 and 1.98. *See also* MPEP § 609 ("An information disclosure statement filed in accordance with the provisions of 37 CFR 1.97 and 37 CFR 1.98 will be considered by the examiner assigned to the application.").

Objections to the Specification

The Action notes that the trademark "Microsoft" and the phrase "Microsoft CLR" appear in the specification. Applicants recognize that "Microsoft" is a trademarked term. In order to clarify that "Microsoft" is a trademarked term, and to clarify the phrase "Microsoft CLR," Applicants have amended the specification to recognize the trademark and include the "generic terminology" for the phrase "Microsoft CLR" as suggested by MPEP § 608.01(v). Specifically, Applicants have amended the specification to include the language "(the Common Language Runtime (CLR) is Microsoft's commercial implementation of the Common Language Infrastructure (CLI) specification; Microsoft is a trademark of Microsoft Corporation)" after the first occurrence of the phrase "Microsoft CLR."

Claim Objections

The Action objects to minor typographical errors in claims 5 and 18. Applicants have amended claims 5 and 18 to correct these minor typographical errors.

112 Rejection

The Action rejects claims 46 and 47 for insufficient antecedent basis. Applicants have amended claims 46 and 47 to clarify antecedent basis.

Cited Art

The Action cites "The Jalapeno Dynamic Optimizing Compiler for Java" by Burke et al. ("Burke"), "Compilers: Principles, Techniques, and Tools" by Aho et al. ("Aho"), "Pure Java 2" by Litwak ("Litwak"), "Marmot: An Optimizing Compiler for Java" by Fitzgerald et al.

("Fitzgerald"), "C/C++ Language Reference: try-except Statement" by Microsoft ("Microsoft Language Reference"), "A single intermediate language that supports multiple implementations of exceptions" by Ramsey et al. ("Ramsey"), U.S. Patent No. 6,421,667 to Codd et al. ("Codd"), U.S. Patent No. 5,918,235 to Kirshenbaum et al. ("Kirshenbaum"), U.S. Patent No. 5,966,702 to Fresko et al. ("Fresko"), and U.S. Patent No. 6,289,446 to Nilsson ("Nilsson").

101 Rejection

The Action rejects claims 1-52 under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Applicants respectfully disagree, but in order to expedite prosecution, have amended the claims as outlined below. The claims, as amended, are not subject to a § 101 rejection.

The Supreme Court has interpreted § 101 broadly, "[a]s the Supreme Court has held, Congress chose the expansive language of 35 U.S.C. 101 so as to include 'anything under the sun that is made by man.' *Diamond v. Chakrabarty*, 447 U.S. 303, 308-09, 206 USPQ 193, 197 (1980)." MPEP § 2106(IV)(A). "Accordingly, a complete definition of the scope of 35 U.S.C. 101, reflecting Congressional intent, is that any new and useful process, machine, manufacture or composition of matter under the sun that is made by man is the proper subject matter of a patent." MPEP § 2106(IV)(A).

Therefore, in order to be statutory under § 101, "[t]he claimed invention as a whole must accomplish a practical application. That is, it must produce a 'useful, concrete and tangible result.' *State Street*, 149 F.3d at 1373, 47 USPQ2d at 1601-02." MPEP § 2106(II)(A).

Furthermore, as the Board of Patent Appeals and Interferences recently held in *Ex parte Lundgren*, there is no separate "technological arts" test to determine patentable subject matter under § 101. *Ex parte Lundgren*, Appeal No. 2003-2088 (BPAI 2005).

Patentability of Claims 1-13 under § 101

The Action rejects claims 1-13 under 35 U.S.C. § 101, stating that the claims might be "directed merely to an abstract idea that is not tied to a technological art, environment or machine which would result in a practical application producing a concrete, useful, and tangible result." Action, page 3.

Applicants have amended claim 1 to clarify that claim 1 is directed to "A method *implemented at least in part by a computing device*" and that the method comprises "generating, *in a computer-readable media having a tangible component*, a computer-readable version." Support for the amendment can be found, for example, in the Application at page 37, line 25 to page 38, line 14.

Because claim 1 is directed to "A method *implemented at least in part by a computing device*" and that the method comprises "generating, *in a computer-readable media having a tangible component*, a computer-readable version," claim 1 results in a practical application producing a concrete, useful, and tangible result. Therefore, claim 1, and dependent claims 2-13, should not be subject to a § 101 rejection.

Patentability of Claims 15-23 and 25-31 under § 101

The Action rejects claims 15-23 and 25-31 under 35 U.S.C. § 101, stating that the "system does not appear to be a tangible embodiment of a system," and that the claim limitations are not "necessarily implemented in hardware." Action, pages 3-4.

Applicants have amended claim 15 to clarify that it comprises "an intermediate language reader, *implemented at least in part by a computing device of the system*, for obtaining an intermediate language representation of a source code file and generating a source language independent intermediate representation of exception handling constructs of the source code based on the intermediate language representation." Support for the amendment can be found, for example, in the Application at page 37, line 25 to page 38, line 14.

Because claim 15 is directed to "A system ... comprising: an intermediate language reader, *implemented at least in part by a computing device of the system*, for obtaining an intermediate language representation of a source code file and generating a source language independent intermediate representation of exception handling constructs of the source code based on the intermediate language representation ...," claim 15 results in a practical application producing a concrete, useful, and tangible result. Therefore, claim 15, and dependent claims 16-23 and 25-31, should not be subject to a § 101 rejection.

Patentability of Claims 32-52 under § 101

The Action rejects claims 32-52 under 35 U.S.C. § 101, stating that the claimed computer readable storage medium "allows non-tangible interpretations of the medium such as a carrier wave medium." Action, page 4.

Applicants have amended claims 32, 44, 48, and 50 to clarify that they are directed to a "computer readable storage medium *having a tangible component.*" Support for the amendment can be found, for example, in the Application at page 37, line 25 to page 38, line 14.

Because claims 32, 44, 48, and 50 are directed to a "computer readable storage medium *having a tangible component,*" claims 32, 44, 48, and 50 results in a practical application producing a concrete, useful, and tangible result. Therefore, claims 32, 44, 48, and 50, and associated dependent claims 33-43, 45-47, 49, and 51-52, should not be subject to a § 101 rejection.

102 Rejection

Patentability of Claims 1, 15, 16, and 53 over Burke under § 102(b)

The Action rejects claims 1, 15, 16, and 53 under 35 U.S.C. § 102(b) as being anticipated by Burke. Applicants respectfully submit that the claims in their present form are allowable over the cited art. To establish a prima facie case of anticipation, the cited art must show each and every element as set forth in a claim. MPEP § 2131.01.

Claim 1

Claim 1 has been amended, in part, to clarify that it is directed to "A method ... of processing a source language independent intermediate representation of software comprising exception handling constructs." Amended claim 1 reads as follows (emphasis added):

> A method implemented at least in part by a computing device of processing a *source language independent intermediate representation* of software comprising exception handling constructs, the method comprising:
>
> reading the *source language independent intermediate representation* of software comprising exception handling constructs; wherein the *source language independent intermediate representation* explicitly expresses exception handling control flow of the software; and
>
> generating, in a computer-readable media having a tangible component, a computer-readable version of the software implementing the exception handling

control flow based on the *source language independent intermediate representation.*

Burke's description of a Java optimizing compiler does not anticipate the "source language independent intermediate representation of software comprising exception handling constructs" of claim 1. For example, the Application at p. 9, lines 2-17, and Figs. 2 and 3A, describes the source language independent intermediate representation as follows:

> Language independent intermediate representation of exception handling constructs
>
> FIG. 2 illustrates a system 200 for implementing a uniform exception handling intermediate representation 230 for multiple source languages (205-208) for code optimization by the compiler back end 240. As shown in FIG. 2, the system 200 includes a intermediate language (IL) representation 210-213 for each of the multiple source code representations 205-208 which is parsed or read by an IL reader 220 which translates the multiple IL representations 210-213 to a single intermediate representation 230. The IL representation is a higher-level intermediate representation than the intermediate representation 230 and may be expressed in any number of well known intermediate languages such as MSIL (Microsoft CLR) (for C#, Visual Basic, JScript, C, and FORTRAN) and CIL (for C++). Even though the system 200 for generating a uniform exception handling framework for multiple languages is shown as having a single IL reader process for multiple source languages, it is possible to implement multiple such readers, each corresponding to one or more of the IL representations 210-213.

Burke describes that "[t]he Jalapeño Dynamic Optimizing Compiler is a key component of the Jalapeño Virtual Machine, a new Java Virtual Machine (JVM) designed to support efficient and scalable execution of Java applications on SMP server machines." Burke Abstract. In describing the Jalapeño Optimizing Compiler, Burke describes, "[i]n addition, its optimizations must deliver significant performance improvements while correctly preserving Java semantics with respect to exceptions, garbage collection, and threads." Burke p. 130, section 3. Thus, while Burke describes "[a] key difference between the Jalapeño HIR and Java bytecodes is the addition of separate operators to implement explicit checks for several common run-time exceptions ...," the intermediate representation described by Burke is specific to the Java programming language. Burke p. 131, section 4.

Applicants recognized the problem of having an intermediate representation tied to a specific programming language:

> Also, traditionally, intermediate representations have been specific to a source language. Thus, compilers have to be aware of the specific exception handling models of the source language associated with each representation.
> ...
> Thus, there is a need for a single uniform framework for intermediately representing exception handling constructs that is uniform across multiple models for representing exception handling and is capable of accounting for the various attributes of such models described above. Application p. 2, lines 17-19, and p. 3, line 28 to p. 4, line 3.

Because Burke does not describe a "source language independent intermediate representation" as recited by claim 1, Burke does not anticipate claim 1. Therefore, claim 1 should be in condition for allowance.

Furthermore, Burke's description of exception checking in a Java optimizing compiler does not anticipate "wherein the source language independent intermediate representation explicitly expresses exception handling control flow of the software" as recited by claim 1. For example, the Application at p. 11, line 10, to p. 12, line 6, and Figs. 8 and 12, describes explicitly expressing exception handling control flow as follows:

> Exception causing instructions explicitly expressed within the main control flow of the intermediate representation
>
> Exception causing instructions are guarded by their handlers or finally regions. When an instruction causes an exception the control flow may pass to a handler and sometimes the handler may be conditionally selected based on the processing of filter instructions.
> ...
> FIGS. 8 and 12 illustrate pseudo code representing various exception handling related instructions.
> ...
> As described with reference to FIG. 3, the intermediate language representations with various models for expressing exception handling maybe analyzed to determine control flow between the exception causing instructions and their respective handlers and continuations, which may then be explicitly expressed within the same control flow as the rest of the instructions that do not cause exceptions. One way to accomplish this is to build a control flow representation using instructions with a modest memory allocation cost such as, one word per instruction. The handlers may be represented by instructions that use an exception variable which may be defined by an exception causing instruction. The handler or filter instructions can then test the exception variable and branch to the handler body or to another handler based on the value or type of the exception object.

In addition, the Application at p. 12, lines 14-18, and Fig. 5, describes, for example, a format for instructions implementing the exception handling control flow as follows:

> As noted above, the intermediate representation of exception handling constructs in the intermediate language representation may be expressed at an instruction level. FIG. 5 shows one such general implementation of a data structure for instructions or nodes (IR nodes) that will allow the exception handling constructs to be expressed within the control flow of the intermediate representation of the rest of the code.

Burke describes the "Jalapeño HIR" as including the "addition of separate operators to implement explicit checks for several common run-time exceptions, e.g., NULL_CHECK and BOUNDS_CHECK operators to test for null pointer dereferences and out-of-bounds array accesses respectively." Burke p. 131, section 4. Burke illustrates the HIR exception checking instructions at p. 132-133, section 5.1, and Fig. 6. As understood by applicants, while Burke does include exception checking instructions (e.g., the null_check instruction), Burke does not explicitly express exception handling control flow as recited by claim 1 because in Burke there is no explicit representation of the control flow for the exception checks in Burke's intermediate representation (HIR).

As can be seen, for example, at page 20 and Fig. 13 of the Application, exception handling control flow is explicitly expressed in the source language independent intermediate representation. Because Burke does not explicitly express exception handling control flow in the intermediate representation as recited by claim 1, Burke does not anticipate claim 1.

Claim 15

Claim 15 has been amended, in part, to clarify that it is directed to "A system for implementing source language independent exception handling intermediate representations for multiple source code languages." Amended claim 15 reads as follows (emphasis added):

> A system for implementing *source language independent exception handling intermediate representations for multiple source code languages*, the system comprising:
>
> an intermediate language reader, implemented at least in part by a computing device of the system, for obtaining an intermediate language representation of a source code file and generating a *source language independent intermediate representation* of exception handling constructs of the source code based on the intermediate language representation;

wherein the *source language independent intermediate representation* explicitly expresses exception handling control flow of the source code.

As discussed with regard to claim 1 above, Burke's description of a Java optimizing compiler does not anticipate the "source language independent intermediate representation" and "wherein the source language independent intermediate representation explicitly expresses exception handling control flow of the source code" language of claim 15. Therefore, claim 15 should be in condition for allowance.

Claim 16

Claim 16 depends on claim 15. Thus, at least for the reasons set forth above with regard to claim 15, claim 16 should be in condition for allowance.

Claim 53

Claim 53 reads as follows (emphasis added):

> A system for implementing *uniform exception handling intermediate representations for multiple source code languages*, the system comprising:
> means for reading an intermediate language representation of a source code file and generating a *uniform intermediate representation* of exception handling constructs of the source code based on the intermediate language representation;
> wherein the *uniform intermediate representation explicitly expresses exception handling control flow of the source code.*

Burke's description of exception checking in a Java optimizing compiler does not anticipate "wherein the uniform intermediate representation explicitly expresses exception handling control flow of the source code" as recited by claim 53. For example, the Application at p. 11, line 10, to p. 12, line 6, and Figs. 8 and 12, describes explicitly expressing exception handling control flow as follows:

> Exception causing instructions explicitly expressed within the main control flow of the intermediate representation
>
> Exception causing instructions are guarded by their handlers or finally regions. When an instruction causes an exception the control flow may pass to a handler and sometimes the handler may be conditionally selected based on the processing of filter instructions.

> ...
> FIGS. 8 and 12 illustrate pseudo code representing various exception handling related instructions.
> ...
> As described with reference to FIG. 3, the intermediate language representations with various models for expressing exception handling maybe analyzed to determine control flow between the exception causing instructions and their respective handlers and continuations, which may then be explicitly expressed within the same control flow as the rest of the instructions that do not cause exceptions. One way to accomplish this is to build a control flow representation using instructions with a modest memory allocation cost such as, one word per instruction. The handlers may be represented by instructions that use an exception variable which may be defined by an exception causing instruction. The handler or filter instructions can then test the exception variable and branch to the handler body or to another handler based on the value or type of the exception object.

In addition, the Application at p. 12, lines 14-18, and Fig. 5, describes, for example, a format for instructions implementing the exception handling control flow as follows:

> As noted above, the intermediate representation of exception handling constructs in the intermediate language representation may be expressed at an instruction level. FIG. 5 shows one such general implementation of a data structure for instructions or nodes (IR nodes) that will allow the exception handling constructs to be expressed within the control flow of the intermediate representation of the rest of the code.

Burke describes the "Jalapeño HIR" as including the "addition of separate operators to implement explicit checks for several common run-time exceptions, e.g., NULL_CHECK and BOUNDS_CHECK operators to test for null pointer dereferences and out-of-bounds array accesses respectively." Burke p. 131, section 4. Burke illustrates the HIR exception checking instructions at p. 132-133, section 5.1, and Fig. 6. As understood by applicants, while Burke does include exception checking instructions (e.g., the null_check instruction), Burke does not explicitly express exception handling control flow as recited by claim 53 because in Burke there is no explicit representation of the control flow for the exception checks in Burke's intermediate representation (HIR).

As can be seen, for example, at page 20 and Fig. 13 of the Application, exception handling control flow is explicitly expressed in the source language independent intermediate representation. Because Burke does not explicitly express exception handling control flow in the intermediate representation as recited by claim 53, Burke does not anticipate claim 53.

Furthermore, Burke's description of a Java optimizing compiler does not anticipate "uniform exception handling intermediate representations for multiple source code languages" as recited by claim 53. Burke describes that "[t]he Jalapeño Dynamic Optimizing Compiler is a key component of the Jalapeño Virtual Machine, a new Java Virtual Machine (JVM) designed to support efficient and scalable execution of Java applications on SMP server machines." Burke Abstract. In describing the Jalapeño Optimizing Compiler, Burke describes, "[i]n addition, its optimizations must deliver significant performance improvements while correctly preserving Java semantics with respect to exceptions, garbage collection, and threads." Burke p. 130, section 3. Thus, while Burke describes "[a] key difference between the Jalapeño HIR and Java bytecodes is the addition of separate operators to implement explicit checks for several common run-time exceptions ...," the intermediate representation described by Burke is specific to the Java programming language. Burke p. 131, section 4.

Because Burke does not describe "uniform exception handling intermediate representations for multiple source code languages" as recited by claim 53, Burke does not anticipate claim 53. Therefore, claim 53 should be in condition for allowance.

103 Rejection

Patentability of Claims 2, 5-7, 17, 18, 21, 22, 32, 33-36, 39 and 40 over Burke, Aho, Litwak, Fitzgerald, and Codd under § 103(a)

The Action rejected claims 2, 5-7, 17, 18, 21, 22, 32, 33-36, 39 and 40 under 35 U.S.C. § 103(a) as unpatentable over Burke, Aho, Litwak, Fitzgerald, and Codd. Applicants respectfully submit the claims in their present form are allowable over the cited art.

To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. MPEP § 2142. Motivations to combine or modify references must come from the references themselves or be within the body of knowledge in the art. MPEP § 2143.01.

Claims 2 and 5-7

Claims 2 and 5-7 ultimately depend from claim 1. Therefore, for at least the reasons stated above with regard to claim 1, the Action's § 103(a) rejection of claims 2 and 5-7 also cannot be supported. Thus, the claims are in condition for allowance.

Claims 17, 18, 21, and 22

Claims 17, 18, 21, and 22 ultimately depend from claim 15. Therefore, for at least the reasons stated above with regard to claim 15, the Action's § 103(a) rejection of claims 17, 18, 21, and 22 also cannot be supported. Thus, the claims are in condition for allowance.

Claim 32

Claim 32 has been amended to clarify that is directed to "A computer readable storage medium having a tangible component, and having stored thereon a source language independent intermediate representation of exception handling constructs of source code, the source language independent intermediate representation of exception handling constructs comprising:." Claim 32 reads as follows (emphasis added):

> A computer readable storage medium having a tangible component, and having stored thereon a *source language independent intermediate representation* of exception handling constructs of source code, the *source language independent intermediate representation* of exception handling constructs comprising:
> a first instruction for expressing explicit transfer of control to a finalization code block;
> a second instruction for expressing acceptance of control transfer into the finalization code block; and
> a third instruction for expressing transfer of control out of the finalization code block.

For example, the Application at p. 9, lines 2-17, and Figs. 2 and 3A, describes the source language independent intermediate representation as follows:

> Language independent intermediate representation of exception handling constructs
>
> FIG. 2 illustrates a system 200 for implementing a uniform exception handling intermediate representation 230 for multiple source languages (205-208) for code optimization by the compiler back end 240. As shown in FIG. 2, the system 200 includes a intermediate language (IL) representation 210-213 for each

> of the multiple source code representations 205-208 which is parsed or read by an IL reader 220 which translates the multiple IL representations 210-213 to a single intermediate representation 230. The IL representation is a higher-level intermediate representation than the intermediate representation 230 and may be expressed in any number of well known intermediate languages such as MSIL (Microsoft CLR) (for C#, Visual Basic, JScript, C, and FORTRAN) and CIL (for C++). Even though the system 200 for generating a uniform exception handling framework for multiple languages is shown as having a single IL reader process for multiple source languages, it is possible to implement multiple such readers, each corresponding to one or more of the IL representations 210-213.

As discussed above with regard to claim 1, Burke does not describe a "source language independent intermediate representation" as recited by claim 32. In addition, as understood by Applicants, neither Aho, Litwak, Fitzgerald, or Codd add sufficient disclosure to teach or suggest this language. Therefore, claim 32 should be in condition for allowance.

Claims 33-36, 39 and 40

Claims 33-36, 39 and 40 ultimately depend from claim 32. Therefore, for at least the reasons stated above with regard to claim 32, the Action's § 103(a) rejection of claims 33-36, 39 and 40 also cannot be supported. Thus, the claims are in condition for allowance.

Patentability of Claims 3, 4, 19, 20, 41, and 42 over Burke, Aho, Litwak, Fitzgerald, Codd, and Kirshenbaum under § 103(a)

The Action rejected claims 3, 4, 19, 20, 41, and 42 under 35 U.S.C. § 103(a) as unpatentable over Burke, Aho, Litwak, Fitzgerald, Codd, and Kirshenbaum. Applicants respectfully submit the claims in their present form are allowable over the cited art.

Claims 3 and 4

Claims 3 and 4 ultimately depend from claim 1. Therefore, for at least the reasons stated above with regard to claim 1, the Action's § 103(a) rejection of claims 3 and 4 also cannot be supported. Thus, the claims are in condition for allowance.

Claims 19 and 20

Claims 19 and 20 ultimately depend from claim 15. Therefore, for at least the reasons stated above with regard to claim 15, the Action's § 103(a) rejection of claims 19 and 20 also cannot be supported. Thus, the claims are in condition for allowance.

Claims 41 and 42

Claims 41 and 42 depend from claim 32. Therefore, for at least the reasons stated above with regard to claim 32, the Action's § 103(a) rejection of claims 41 and 42 also cannot be supported. Thus, the claims are in condition for allowance.

Patentability of Claims 8, 23, 37, and 38 over Burke, Aho, Litwak, Fitzgerald, Codd, and Fresko under § 103(a)

The Action rejected claims 8, 23, 37, and 38 under 35 U.S.C. § 103(a) as unpatentable over Burke, Aho, Litwak, Fitzgerald, Codd, and Fresko. Applicants respectfully submit the claims in their present form are allowable over the cited art.

Claim 8

Claim 8 ultimately depends from claim 1. Therefore, for at least the reasons stated above with regard to claim 1, the Action's § 103(a) rejection of claim 8 also cannot be supported. Thus, the claim is in condition for allowance.

Claim 23

Claim 23 has been amended to include language from dependent claim 24. Claim 23 ultimately depends from claim 15. Therefore, for at least the reasons stated above with regard to claim 15, the Action's § 103(a) rejection of claim 23 also cannot be supported. Thus, the claim is in condition for allowance.

Claims 37 and 38

Claims 37 and 38 ultimately depend from claim 32. Therefore, for at least the reasons stated above with regard to claim 32, the Action's § 103(a) rejection of claims 37 and 38 also cannot be supported. Thus, the claims are in condition for allowance.

Patentability of Claims 9, 10, 25, and 26 over Burke, Aho, and Litwak under § 103(a)

The Action rejected claims 9, 10, 25, and 26 under 35 U.S.C. § 103(a) as unpatentable over Burke, Aho, and Litwak. Applicants respectfully submit the claims in their present form are allowable over the cited art.

Claims 9 and 10

Claims 9 and 10 ultimately depend from claim 1. Therefore, for at least the reasons stated above with regard to claim 1, the Action's § 103(a) rejection of claims 9 and 10 also cannot be supported. Thus, the claims are in condition for allowance.

Claims 25 and 26

Claims 25 and 26 ultimately depend from claim 15. Therefore, for at least the reasons stated above with regard to claim 15, the Action's § 103(a) rejection of claims 25 and 26 also cannot be supported. Thus, the claims are in condition for allowance.

Patentability of Claims 11, 12, 27, 28, 48, and 49 over Burke, Aho, Litwak, and Fresko under § 103(a)

The Action rejected claims 11, 12, 27, 28, 48, and 49 under 35 U.S.C. § 103(a) as unpatentable over Burke, Aho, Litwak, and Fresko. Applicants respectfully submit the claims in their present form are allowable over the cited art.

Claims 11 and 12

Claims 11 and 12 ultimately depend from claim 1. Therefore, for at least the reasons stated above with regard to claim 1, the Action's § 103(a) rejection of claims 11 and 12 also cannot be supported. Thus, the claims are in condition for allowance.

Claims 27 and 28

Claims 27 and 28 ultimately depend from claim 15. Therefore, for at least the reasons stated above with regard to claim 15, the Action's § 103(a) rejection of claims 27 and 28 also cannot be supported. Thus, the claims are in condition for allowance.

Claim 48

Claim 48 has been amended to clarify that is directed to "A computer readable storage medium having a tangible component, and having stored thereon a source language independent intermediate representation of exception handling constructs of source code, the source language independent intermediate representation of exception handling constructs comprising:." Claim 48 reads as follows (emphasis added):

> A computer readable storage medium having a tangible component, and having stored thereon a *source language independent intermediate representation* of exception handling constructs of source code, the *source language independent intermediate representation* of exception handling constructs comprising:
> an instruction for specifying a handler for an exception based on a type value of an exception object related to the exception, wherein a destination operand of the instruction comprises a predetermined exception object, a first source operand of the instruction comprises a label indicative of a code block related to the handler and second source operand comprises a label indicative of a code block related to a continuation.

For example, the Application at p. 9, lines 2-17, and Figs. 2 and 3A, describes the source language independent intermediate representation.

As discussed above with regard to claim 1, Burke does not describe a "source language independent intermediate representation" as recited by claim 48. In addition, as understood by Applicants, neither Aho, Litwak, or Fresko add sufficient disclosure to teach or suggest this language. Therefore, claim 48 should be in condition for allowance.

Claim 49

Claim 49 depends from claim 48. Therefore, for at least the reasons stated above with regard to claim 48, the Action's § 103(a) rejection of claim 49 also cannot be supported. Thus, the claim is in condition for allowance.

Patentability of Claims 13, 29-31, and 50-52 over Burke in view of Microsoft Language Reference under § 103(a)

The Action rejected claims 13, 29-31, and 50-52 under 35 U.S.C. § 103(a) as unpatentable over Burke in view of Microsoft Language Reference. Applicants respectfully submit the claims in their present form are allowable over the cited art.

Claim 13

Claim 13 has been amended to include language from dependent claim 14. Claim 13 depends from claim 1. Therefore, for at least the reasons stated above with regard to claim 1, the Action's § 103(a) rejection of claim 13 also cannot be supported. Thus, the claim is in condition for allowance.

Claims 29-31

Claims 29-31 ultimately depend from claim 15. Therefore, for at least the reasons stated above with regard to claim 15, the Action's § 103(a) rejection of claims 29-31 also cannot be supported. Thus, the claims are in condition for allowance.

Claim 50

Claim 50 has been amended to clarify that is directed to "A computer readable storage medium having a tangible component, and having stored thereon a source language independent intermediate representation of exception handling constructs of source code, the source language independent intermediate representation of exception handling constructs comprising:." Claim 50 reads as follows (emphasis added):

> A computer readable storage medium having a tangible component, and having stored thereon a *source language independent intermediate representation* of exception handling constructs of source code, the *source language independent intermediate representation* of exception handling constructs comprising:
>
> a first instruction for indicating entry into a try-except region; and
>
> a second instruction for selecting one of a plurality of control flow paths for exception handling based on a type value related to the exception, wherein the plurality of control flow paths available for selection includes a path related to resumption of execution of an instruction causing the exception.

For example, the Application at p. 9, lines 2-17, and Figs. 2 and 3A, describes the source language independent intermediate representation

As discussed above with regard to claim 1, Burke does not describe a "source language independent intermediate representation" as recited by claim 50. In addition, as understood by Applicants, Microsoft Language Reference does not add sufficient disclosure to teach or suggest this language. Therefore, claim 50 should be in condition for allowance.

Claims 51 and 52

Claims 51 and 52 depend from claim 50. Therefore, for at least the reasons stated above with regard to claim 50, the Action's § 103(a) rejection of claims 51 and 52 also cannot be supported. Thus, the claims are in condition for allowance.

Patentability of Claim 43 over Burke, Aho, Litwak, Fitzgerald, Codd, and Nilsson under § 103(a)

The Action rejected claim 43 under 35 U.S.C. § 103(a) as unpatentable over Burke, Aho, Litwak, Fitzgerald, Codd, and Nilsson. Applicants respectfully submit the claim in its present form is allowable over the cited art.

Claim 43

Claim 43 ultimately depends from claim 32. Therefore, for at least the reasons stated above with regard to claim 32, the Action's § 103(a) rejection of claim 43 also cannot be supported. Thus, the claim is in condition for allowance.

Patentability of Claims 44-46 over Burke, Aho, Litwak, and Codd under § 103(a)

The Action rejected claims 44-46 under 35 U.S.C. § 103(a) as unpatentable over Burke, Aho, Litwak, and Codd. Applicants respectfully submit the claims in their present form are allowable over the cited art.

Claim 44

Claim 44 has been amended to clarify that is directed to "A computer readable storage medium having a tangible component, and having stored thereon a source language independent intermediate representation of exception handling constructs of source code, the source language independent intermediate representation of exception handling constructs comprising:." Claim 44 reads as follows (emphasis added):

> A computer readable storage medium having a tangible component, and having stored thereon a *source language independent intermediate representation* of exception handling constructs of source code, the *source language independent*

intermediate representation of exception handling constructs comprising:

a first instruction for catching an exception and returning an exception object related to the exception; and

a second instruction for specifying a handler for the exception based on a type value of the exception object.

For example, the Application at p. 9, lines 2-17, and Figs. 2 and 3A, describes the source language independent intermediate representation.

As discussed above with regard to claim 1, Burke does not describe a "source language independent intermediate representation" as recited by claim 44. In addition, as understood by Applicants, neither Aho, Litwak, or Codd add sufficient disclosure to teach or suggest this language. Therefore, claim 44 should be in condition for allowance.

Claims 45 and 46

Claims 45 and 46 ultimately depend from claim 44. Therefore, for at least the reasons stated above with regard to claim 44, the Action's § 103(a) rejection of claims 45 and 46 also cannot be supported. Thus, the claims are in condition for allowance.

Patentability of Claim 47 over Burke, Aho, Litwak, Codd, and Ramsey under § 103(a)

The Action rejected claim 47 under 35 U.S.C. § 103(a) as unpatentable over Burke, Aho, Litwak, Codd, and Ramsey. Applicants respectfully submit the claim in its present form is allowable over the cited art.

Claim 47

Claim 47 ultimately depends from claim 44. Therefore, for at least the reasons stated above with regard to claim 44, the Action's § 103(a) rejection of claim 47 also cannot be supported. Thus, the claim is in condition for allowance.

Other Dependent Claim Amendments

Various dependent claims have been amended to correspond with amended language from their respective independent claims.

New Claim 54

Claim 54, which is dependent on claim 1, is directed to claim 1 "wherein the source language independent intermediate representation explicitly expresses exception handling control flow of the software by associating exception causing instructions with labels representing their related handlers, and wherein each instruction of the source language independent intermediate representation comprises a handler field." Support for claim 54 is found, for example, in the Application at p. 11-13 and Fig. 5. For at least the reasons stated above with regard to claim 1, claim 54 should be in condition for allowance.

New Claim 55

Claim 55, which is dependent on claim 15, is directed to the "generating" language of claim 15. Support for claim 55 is found, for example, in the Application at p. 23-25, and Figs. 12, 13, 18, 19A-C, 20, and 21. For at least for the reasons set forth above with regard to claim 15, claim 55 should be in condition for allowance.

Request for Interview

If any issues remain, the Examiner is formally requested to contact the undersigned attorney prior to issuance of the next Office Action in order to arrange a telephonic interview. It is believed that a brief discussion of the merits of the present application may expedite prosecution. Applicants submit the foregoing formal Amendment so that the Examiner may fully evaluate Applicants' position, thereby enabling the interview to be more focused.

This request is being submitted under MPEP § 713.01, which indicates that an interview may be arranged in advance by a written request.

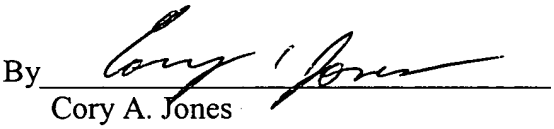
Conclusion

The claims in their present form should now be allowable. Such action is respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Telephone: (503) 595-5300
Facsimile: (503) 595-5301

By 
Cory A. Jones
Registration No. 55,307

CLIENT/MATTER NO. 3382-65591-01 ATTY/SEC CJ/lcm

INVENTOR(S): Grover

APP. NO. 10/609,275 FILING DATE 6-26-03

The following, due _____, mailed 7-14-04

by First Class Mail, was received in the U.S. PTO on the date stamped hereon:

___Amendment/Response ___After Final Action ___ Extension of Time/Extension Fee for ___ Months ___ Revised Drawings ___ Sht(s)	
___ Notice to File Missing Parts - Date _____ ___ Notice to File Corrected App. Papers - Date _____ ___ Comb. Dec./POA ___ Sht(s) Drawings ___ Late Fee	___ Ltr to Draftsperson ___ Notice-Allowability ___ Sht(s) Drawings
___ Issue Fee Transmittal (Form PTOL-85b) ___ Supp. Dec. ___ Issue/Adv. Order fees	___ Sequence Listing ___ pages ___ Stmt in Comp. ___ Disk
X IDS X Form PTO-1449/Ref.	___ Petition
___ Assignment to: _____	___ Recordal Sheet
X Transmittal Ltr. + X 1 ___2 copy(ies)	___Request for Corrected Filing Receipt
___ Check No. _____ for $_____	X Deposit Acct. 02-4550 Authority
___	

OIPE JC15

JUL 16 2004

PATENT & TRADEMARK OFFICE

C

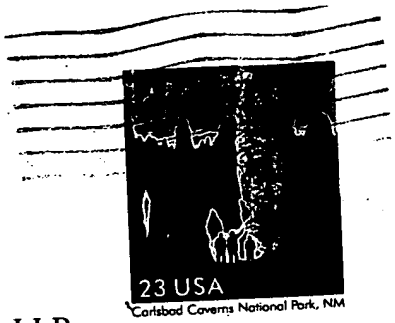
BEST AVAILABLE COPY

RECEIVED

JUL 19 2004

NO VA P&DC 220 PM 15 JUL 2004

KLARQUIST SPARKMAN LLP



KLARQUIST SPARKMAN, LLP
Intellectual Property and Patent Law
One World Trade Center
121 S.W. Salmon St., Suite 1600
Portland, OR 97204-2988

EXHIBIT

tabbies

A